□FIG. 1□
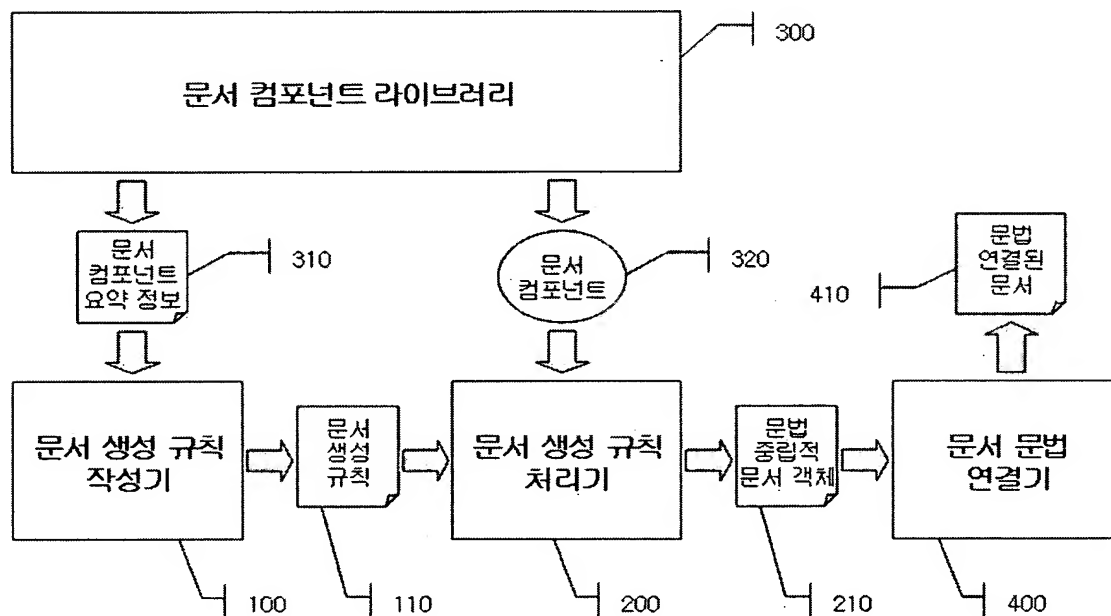
FIG. 1 shows a schematic diagram of a configuration of a component-based automatic document generation system according to an exemplary embodiment of the present invention.

5

300: Document component library

310: Document component summary information

320: Document component

10   410: Grammar connected document

100: Document generation rule formulator
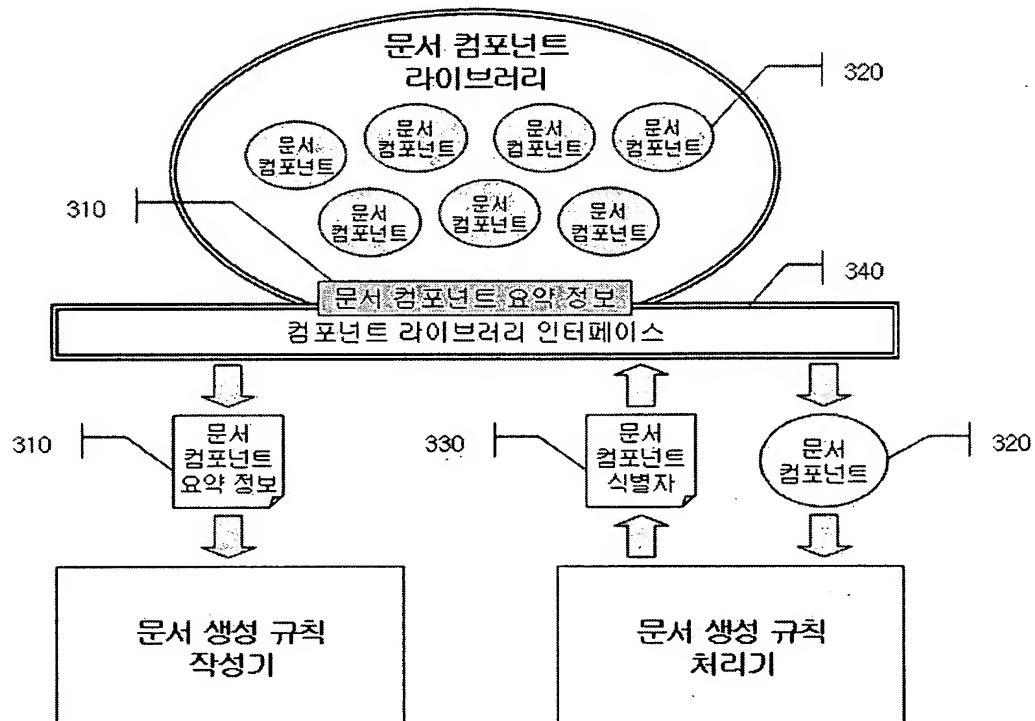
110: Document generation rule

200: Document generation rule processor

16

210: Grammar neutral document

400: Document grammar connector

□FIG. 2□

FIG. 2 shows a diagram of a configuration of a document component library shown in FIG. 1.

5　320: Document component

□□ □□□□ □□□□□: Document component library

310: Document component summary information
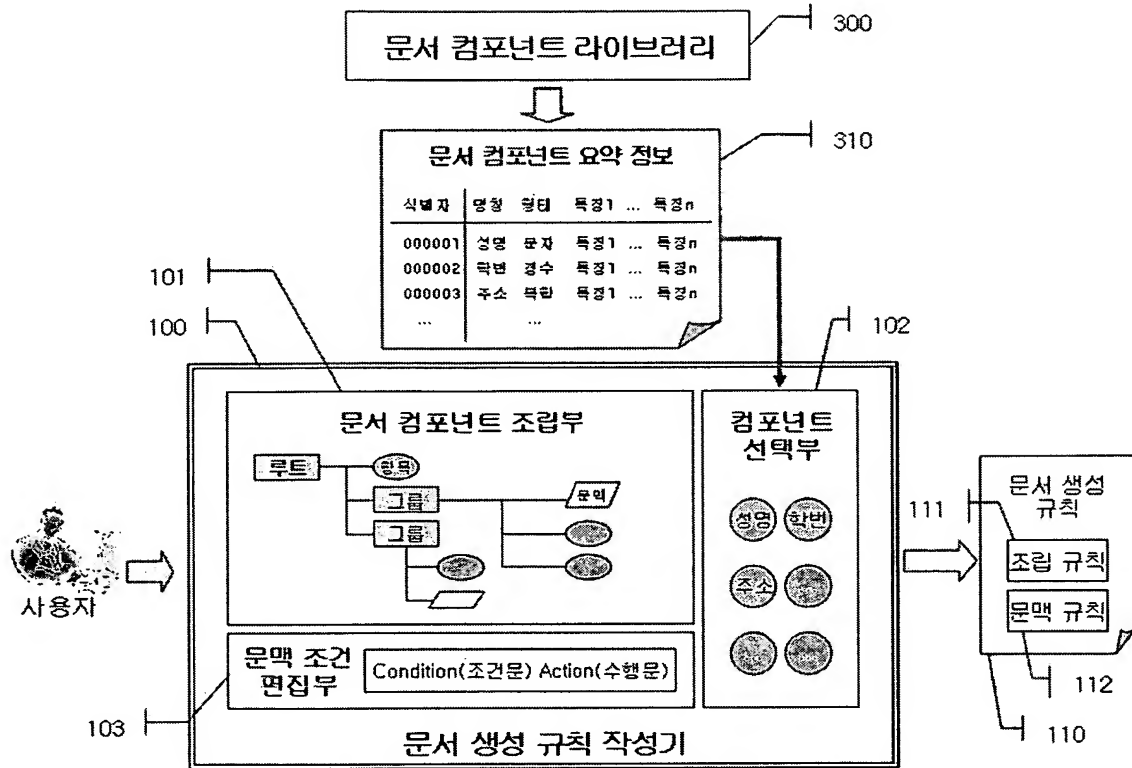
340: Component library interface

330: Document component ID

10　문서 생성 규칙 작성기: Document generation rule formulator

□□ □□ □□ □□□: Document generation rule processor

FIG. 3 shows a diagram of a configuration of a document generation rule formulator shown in FIG. 1.

5   100: Document generation rule formulator

300: Document component library

310: Document component summary information

□□□: ID

□□: Name

10   □□: Type

□□1…□□n: Characteristic 1 … Characteristic n

101: Document component assembler

루트: Root

항목: Item

그룹: Group

문맥: Context

102: Component selector

5 　성명: Name

학번: Student ID number

주소: Address

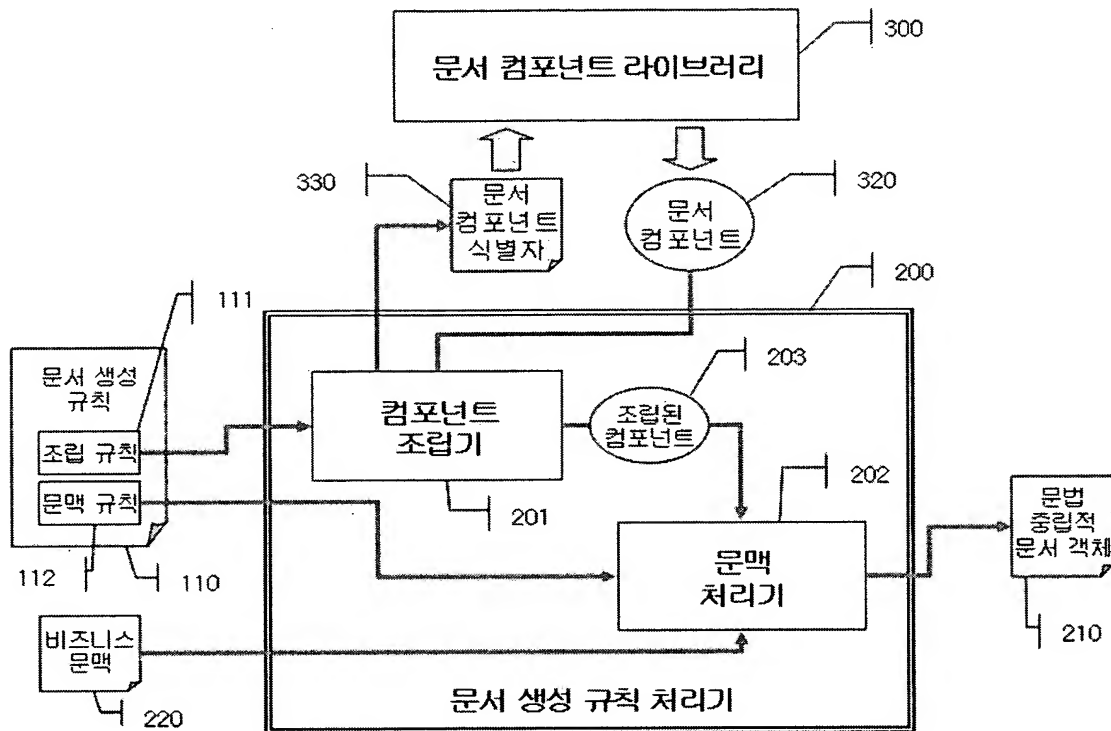103: Context condition compiler

사용자: User

10 　110: Document generation rule

111: Assembly rule

112: Context rule

□FIG. 4□

FIG. 4 shows a diagram of a configuration of a document generation rule processor shown in FIG. 1.

5   300: Document component library

330: Document component ID

320: Document component

110: Document generation rule

111: Assembly rule

10   112: Context rule

200: Document generation rule processor

201: Component assembler
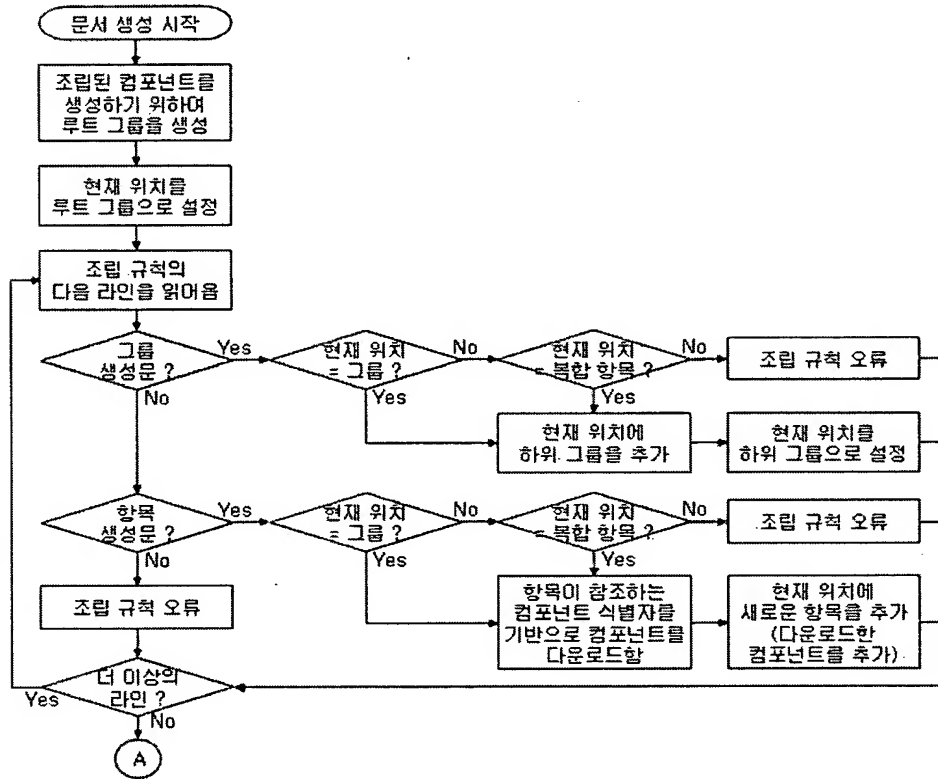
203: Assembled component

202: Context processor

210: Grammar neutral document object

220: Business context

□FIG. 5□

FIG. 5 shows a flowchart representing an operation for processing an assembly rule in a component assembler shown in FIG. 4.

5   □□ □□ □□: Start generating document

□□□ □□□□□ □□□□ □□□ □□□□□ □□: Generate root group to generate assembled component

□□ □□□ □□ □□□□ □□: Establish current location as root group

□□ □□□ □□ □□□ □□□: Read subsequent line of assembly rule

10   □□ □□□□?: Group generation sentence?

□□□□□=□□?: Current location=Group?

□□□□□=□□ □□?: Current location = Complex item?

□□ □□ □□: Assembly rule error

23

□□ □□□ □□ □□□ □□: Add lower group to current location

□□ □□□ □□ □□□□ □□: Establish current location as lower group

□□ □□□?: Item generation sentence?

□□□ □□□□ □□□□ □□□□ □□□□ □□□□□ □□□□: Download component based
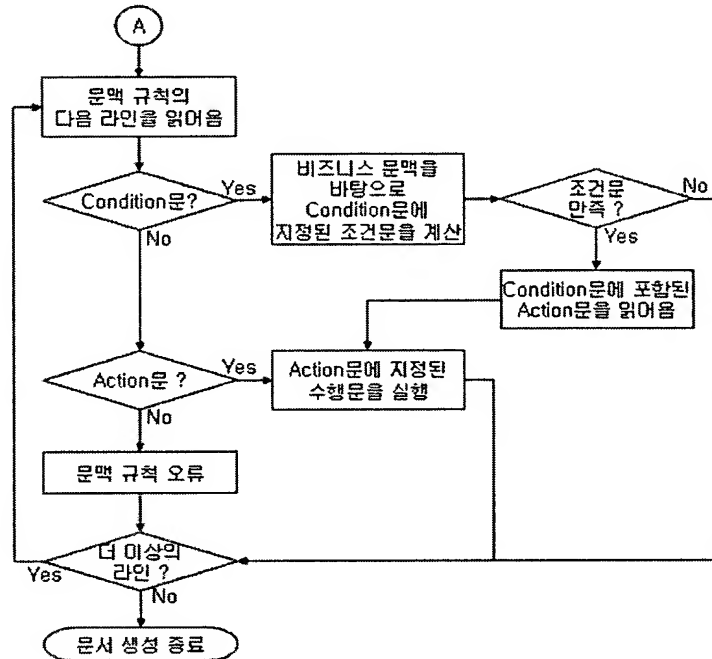
on component ID as reference of item.

□□ □□□ □□□ □□□ □□(□□□□□ □□□□□ □□): Add new item to current location

(Add downloaded component)

□ □□□ □□?: More line?

24

□FIG. 6□

FIG. 6 shows a flowchart representing an operation for processing a context rule in a context processor shown in FIG. 4.

5   □□□□□ □□□□□ □□□: Read subsequent line of context rule

Condition문?: Condition sentence?

비즈니스 문맥을 바탕으로 condition문에 지정된 조건문을 계산: Calculate condition sentence designated to condition sentence based on business context

조건문 만족?: Satisfy condition sentence?

10   Condition□□ □□□ Action□□ □□□: Read action sentence in condition sentence

Action□□ □□□ □□□□ □□: Execute action sentence
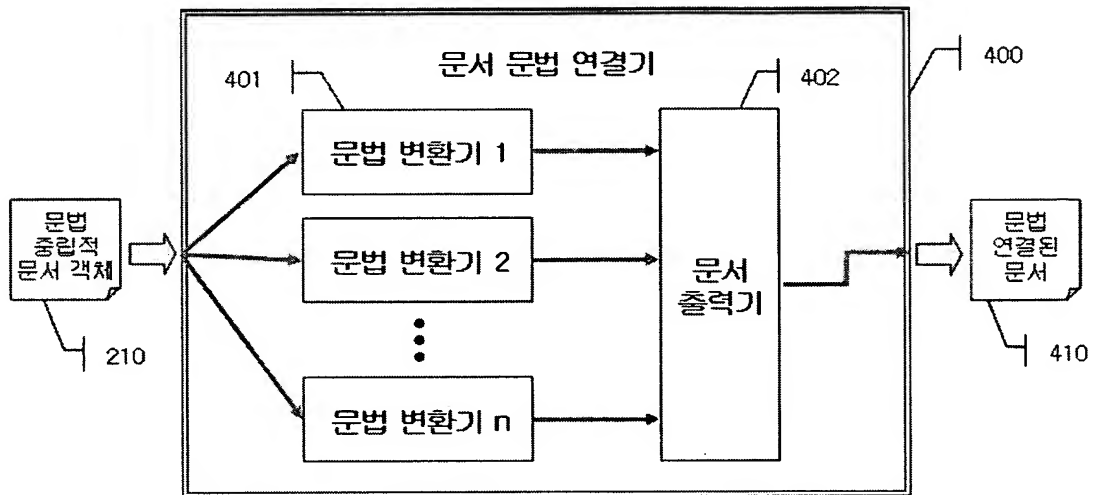
Action□?: Action sentence?

25

⬜⬜ ⬜⬜ ⬜⬜: Context rule error

⬜ ⬜⬜⬜ ⬜⬜: More line?

⬜⬜ ⬜⬜ ⬜⬜: Finish generating document

□FIG. 7□

FIG. 7 shows a diagram of a configuration of a document grammar connector shown in FIG. 1.

5   210: Grammar neutral document

400: Document grammar connector

401: Grammar converter

402: Document output unit

410: Grammar-connected document